3900

Programming System

User Notes — Version 2.5

DATA I/O

984-1013-013

October 1994 984-1013-013

Data I/O has made every attempt to ensure that the information in this document is accurate and complete. Data I/O assumes no liability for errors, or for any incidental, consequential, indirect or special damages, including, without limitation, loss of use, loss or alteration of data, delays, or lost profits or savings, arising from the use of this document or the product which it accompanies.

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose without written permission from Data I/O.

Data I/O Corporation
10525 Willows Road N.E., P.O. Box 97046
Redmond, Washington 98073-9746 USA
(206) 881-6444

Acknowledgments:

Data I/O is a registered trademark of and MatchBook and HiTerm are trademarks of Data I/O Corporation.

Data I/O Corporation acknowledges the trademarks of other organizations for their respective products or services mentioned in this document.

© 1990 – 1994 Data I/O Corporation
All rights reserved

What's New in Version 2.5

The User Notes include the most current information about device support and new features and enhancements to your 3900 Programming System.

Installing Your New Software

Please update your *2900/3900 User Manual* before installing your new version of system software.

After updating your manual, refer to pages 5-40 through 5-44 of the *2900/3900 User Manual* for installation instructions.

Contents of these User Notes

The following sections comprise the remainder of the User Notes:

- **New Features and Enhancements** — A brief description of the changes and additions for this release.
- **Special Notes** — Operational notes and information not found in the User Manual.
- **Mass Storage Module Quick Reference** — Information about using the Mass Storage Module with your programmer.
- **Updated Manual Pages** — Replacement pages that bring your copy of the *2900/3900 User Manual* up-to-date. See the section titled "Updating Your Manual" for more information.

New Features and Enhancements

Version 2.5 adds these new features and changes:

- The programming algorithms for over 300 devices have been updated since the previous release.
- Support for over 150 new devices has been added since the previous release.
- The memory editor and swap data operations now support 32-bit mode. The **Edit Data Word Width** can be set to 4, 8, 16, or 32 bits.
- The following new CRC command was added:

  57] Returns the checksum of the last successful program, load or verify operation. If the last program, load or verify operation was not successful, the error code BC is returned.

- The following new CRC error code was added:

  BC **Last device operation checksum not available.** — This error is returned when no successful load, verify, or program operation has been done since the programmer was last powered up, or when the last load, verify, or program operation was unsuccessful.

- The 3900 now supports the use of the JEDEC U (user data) and E (electrical data) fields. See "JEDEC U and E Fields" on page 4 for more information.
- Keep Current algorithms can have multiple device algorithms per file. See "Keep Current" on page 6 for more information.
- Any device specific information is displayed automatically when a device is selected. See "Footnotes" on page 6 for more details.

Special Notes

To MSM Users Who "Back Rev"

CAUTION: ***To avoid damaging the 3900, if you "back rev" to Version 2.3 after the MSM has been updated to Version 2.4, boot from the Version 2.4 floppy disk and*** not the MSM ***the first time you use Version 2.4 again.***

Device List on a Disk

In response to customer requests, the printed device list was replaced with a device list on a disk. The new device list contains all of the information found in the older printed version and in addition, enables you to generate custom device lists.

You can locate a list of the specific devices you use and optionally print that list or save it to file. In addition, the data in the device list on a disk is formatted to make importing it into a database or spreadsheet easy.

The device list program requires a DOS PC with a high-density 3.5-inch floppy drive.

Note: Do not insert the device list disk in the programmer.

For more information, see the **readme.txt** file, which is included with the device list.

Mass Storage Module Fully Supported

As of Version 2.3, the Mass Storage Module is fully supported on the 3900. You can use the MSM to store data, and you can configure the 3900 to boot from the MSM. (For more information, see the MSM reference sheet included with these User Notes.)

Actel Anti-fuse Devices

As of Version 2.5, the blank check is permanently enabled and verify operations permanently disabled for all Actel anti-fuse (Act1 and Act2) FPGA devices and their second sources.

The **blank check** option and the **number of verify passes** option do not apply to Actel anti-fuse devices. If one of these options is attempted on these devices, the message `Illegal operation for the device selected` is displayed.

Lattice Cross-programming Devices

As of Version 2.5, all RALs have been moved from the **Lattice** manufacturer list to the new **Lattice-XPGM** list.

High-speed Downloading to IBM PS/2.

The IBM PS/2's 16550 advanced UART does not support high-speed download protocol. In order to use high-speed download to a PS/2, do one of the following:

- Configure the PS/2's serial port for standard or "16450" operation (refer to your **PS/2 User Manual** for details).
- Add a standard RS-232C serial card to the PS/2 and dedicate its use to the programmer connection.

JEDEC U and E Fields

AutoSite supports the optional JEDEC U (user data) and E (electrical data) fields. The U and E fields are described below.

Note: Implementation of the JEDEC U and E fields is not part of the JEDEC-3C (JESD3-C) standard.

User Data (U Field)

The U field allows user data fuses that do not affect the logical or electrical functionality of the device to be specified in JEDEC files. For instance, the U field can be used to specify the User Data Signature fuse available in some types of PLD devices because this fuse contains information only (it has no logical or electrical functionality).

The following guidelines apply to the U field:

- The U field must be included for devices with U fuses.
- Each U-field cell must be explicitly provided if the U field is present.
- The F (default fuse state) field does not affect U fuses.
- There can only be one U field in a JEDEC file.
- The U field fuses must be listed in the order they appear in the device.
- The U field must be listed after the L field and E fields (if used), and before the V (test vector) field (if used).
- The U field is specified using binary numbers, since the full number of U-field cells is otherwise unknown.
- The number of cells specified in the U field is not included in the QF (number of fuses) field.
- The U-field cells are not included in the C (fuse checksum) field.
- The U field reads left to right to be consistent with the L (fuse list) and E fields.

The syntax for the U field is as follows:

<User Data Fuse List>::'U'<binary-digit(s)>'*'

The character U begins the U field and is followed by one binary digit for each U fuse. Each binary digit indicates one of two possible states (zero, specifying a low-resistance link, or one, specifying a high-resistance link) for each fuse.

For example,

```
QF24*
L0000
101011000000000000000000*
E10100111*
C011A*
U10110110*
```

The U field must be specified after the L field.

If the E field is used, the U field must be specified after it.

Electrical Data (E field)

The E field allows special feature fuses that do not affect the logic function of the device to be specified in JEDEC files.

The following guidelines apply to the E field:

- The E field must be included for devices with E fuses.
- Each E-field cell must be explicitly provided if the E field is present.
- The F (default fuse state) field does not affect E fuses.
- There can only be one E field in a JEDEC file.
- An E field may appear anywhere in the JEDEC file after the L field, but before the C (checksum) field. If the U field is used, the E field must come before the U (user data) field.
- The E field is specified using binary numbers, since the full number of E-field cells is otherwise unknown.
- The number of cells specified in the E field is not included in the QF (number of fuses) field.
- The E-field cells are included in the C (fuse checksum) field.
- The E field reads left to right for the purpose of checksum calculation.

The syntax for the E field is as follows:

<Electrical Data Fuse List>::'E'<binary digit(s)>'*'

The character E begins the E field and is followed by one binary digit for each E fuse. Each binary digit indicates one of two possible states (zero, specifying a low-resistance link, or one, specifying a high-resistance link) for each fuse. For example,

```
QF24*
L0000                                  The E field must be specified
101011000000000000000000*  ----------after the L field.

E10100111*                             The E field must come before
C011A*  ------------------------------the C field.
U10110110*  ---------------------------If the U field is used, the E field
                                       must be specified before it.
```

Keep Current

As of Version 2.5, Keep Current algorithms can have multiple device algorithms per file.

Devices are still displayed 10 at a time and there is still a limit of 10 devices that can be replaced at any given time, but the new Keep Current algorithm format uses disk space more efficiently and allows greater flexibility than the previous algorithm format.

Note that because the **Delete** function deletes an entire file, you must not delete an algorithm unless you want all algorithms in the file to be deleted.

If using software (such as TaskLink) that uses the **3E]** CRC command to select a device, only the first device in the file can be selected. To select other devices in CRC mode, use the **24D]** command to set the algorithm source to Keep Current, and select by manufacturer and part name with the **33]** and **34]** commands.

Footnotes

When in terminal mode, footnotes are automatically displayed for a selected device. You can disable (or enable) this feature in the **Display Device Footnote** field.

To enable or disable automatic footnote display, start your programmer or handler control software, press [M][C][E][P] to access the Display Device Footnote field, and select **Enable** or **Disable**.

Updating Your Manual

The most current manual pages for Version 2.5 contain the October 1994 printing date at the bottom of the page. Please update your *2900 and 3900 User Manual* with the new pages.

Note: If you updated your manual with the last update (V2.4), you need to replace only those pages that were updated for this version. These pages are shown below in bold type.

Following are brief descriptions of the changes affecting each replacement page:

- xxiii to xxviii — Updated the area code of the Canadian office's phone and fax numbers. Added numbers for new BBS locations.
- 1-5 / 1-6 — Updated the Calibration section.
- 2-13 / 2-14 — Updated to reflect the addition of the Boot disk.
- 2-25 / 2-26 — Added voltage selector problem LED condition.
- 2-29 / 2-30 — Added step for inserting Algorithm/System disk.
- 2-33 / 2-34 — Updated to reflect the addition of the Boot disk.
- 5-39 to 5-46 — Updated the description of the Update procedure to reflect the removal of the security device. Added information for updating the Mass Storage Module, a reminder to format floppy disks on the programmer before using the DOS DISKCOPY command, and a reminder for PROMlink-6 and TaskLink users to select Utilities: Device List Update after updating to a new version of programmer software.
- 5-75 / 5-76 — Clarified the description of the Duplicate Disk command. If you use the DOS DISKCOPY command to make a backup copy of a disk, you must format the destination disk on the programmer BEFORE using DISKCOPY.
- 6-7 / 6-8 — Added the new CRC commands to the summary.
- 6-9 / 6-10 — Updated the description of the hhhhhh; command.
- 6-15 / 6-16 — Added options to the n27] CRC command.
- 6-23 to 6-34 — Added the *n*52], 54], 55], 56], 57], and 58] CRC commands. Added BC error code.
- 7-35 / 7-36 — Added E (electrical data) and U (user data) JEDEC fields.
- 7-38A / 7-38B — Added description for E and U fields.

MSM Quick Reference

This sheet contains general information regarding the Mass Storage Module (MSM) and your programmer. We suggest you put this sheet in the User Manual binder and save it for future reference.

Booting from the MSM

You can boot your programmer directly from the MSM, provided the following conditions are met:

- The MSM has been installed in the programmer
- The system software has been installed on the MSM using the More/Configure/Mass Storage command
- The Boot disk is not in drive A:

Algorithm files are automatically installed on the Mass Storage Module when you update system software. To boot from the Mass Storage Module, you need to install the system files on the MSM. See the Update section of your User Manual for more information.

CAUTION: To avoid damaging the 3900, if you "back rev" to Version 2.3 after the MSM has been updated to Version 2.4, boot from the Version 2.4 floppy disk and not the MSM *the first time you use Version 2.4 again.*

Storage Capacity

The MSM is partitioned into 4 logical drives, with specifications as follows:

Drive	Storage Space	Maximum Number of Files	Data Type
C:	31 MB	512	User data
D:	31 MB	512	User data
H:	7 MB	320	System data
I:	10 MB	320	System data

Storage Suggestions

Drives C: and D: are reserved for user data and drives H: and I: are reserved for system use. Despite the fact that drives H: and I: can be written to and read from, we STRONGLY suggest you use only C: and D: to store your data.

Limitations

Other than the file operations listed below, drives C: and D: on the MSM can be used for all file operations that can be done with a floppy drive:

File Operation	Comments
Format Disk	Format only C: and D:. Formatting H: or I: will render your programmer inoperative until you restore the system software from floppy.
Duplicate Disk	Drives C:, D:, H:, and I: cannot be duplicated using the Duplicate Disk command.

Backing up the MSM

We STRONGLY suggest that you periodically make copies of the files on the C: and D: partitions of the MSM as backups. If the MSM were to "crash," you could restore your data files to C: and D: from your backups, and you could restore the system files from the Boot and Algorithm disks.

3900

Programming System

Replacement Pages — Version 2.5

DATA I/O

Preface

The Preface includes details about contacting Data I/O for technical assistance, repair and warranty services, and Keep Current™ subscription service. The Preface also explains the Bulletin Board Service, typographic conventions and more.

Data I/O Customer Support

United States

For technical assistance, contact

Data I/O Customer Resource Center
Telephone: 800-247-5700
Fax: 206-869-2821

For repair or warranty service, contact

Data I/O Central Dispatch
Telephone: 800-735-6070
Fax: 206-881-0561

For Keep Current subscription service, contact

Data I/O Sales
Telephone: 800-332-8246

Canada

For technical assistance, contact

Data I/O Customer Resource Center
Telephone: 800-247-5700
Fax: 206-869-2821

For repair, warranty service, or Keep Current subscription service, contact

Data I/O Canada
6725 Airport Road, Suite 302
Mississauga, Ontario, L4V 1V2
Telephone: 905-678-0761
Fax: 905-678-7306

United Kingdom

For technical assistance, repair, warranty service, or Keep Current subscription service, contact

Data I/O Limited
660 Eskdale Road
Winnersh, Wokingham
Berkshire RG11 5TS
Telephone: 0734-440011
Fax: 0734-448700

Japan

For technical assistance, repair, warranty service, or Keep Current subscription service, contact

Data I/O Japan
Sumitomoseimei Higashishinbashi Bldg. 8F
2-1-7, Higashi-Shinbashi
Minato-Ku, Tokyo 105
Telephone: 03-3432-6991
Fax: 03-3432-6094 (Sales)
03-3432-6093 (Other)
Telex: 2522685 DATAIO J

Germany

For technical assistance, repair, warranty service, or Keep Current subscription service, contact

Data I/O GmbH
Lochhamer Schlag 5a
D-8032 Gräfelfing
Telephone: 089-858580
Fax: 089-8585810

Other European Countries

For technical assistance, repair, warranty service, or Keep Current subscription service, contact the office below and ask for the number of your local Data I/O representative.

Data I/O Europe
660 Eskdale Road
Winnersh, Wokingham
Berkshire RG11 5TS
United Kingdom
Telephone: 44-0734-448899
Fax: 44-0734-440070

Other Countries Worldwide

For technical assistance, repair, warranty service, or Keep Current subscription service, contact the office below and ask for the number of your local Data I/O representative.

Data I/O Intercontinental
10525 Willows Road N.E.
P.O. Box 97046
Redmond, WA USA 98073-9746
Telephone: 206-881-6444
Fax: 206-882-1043
Telex: 4740166

Technical Assistance

You may contact Data I/O for technical assistance by calling, sending a fax or electronic mail (e-mail), or using the Bulletin Board Service (BBS).

To help us give you quick and accurate assistance, please provide the following information:

- Product version number
- Product serial number (if available)
- Detailed description of the problem you are experiencing
- Error messages (if any)
- Device manufacturer and part number (if device-related)

Calling

Call the appropriate Data I/O Customer Support number listed at the front of the Preface.

When you call, please be at your programmer or computer, have the product manual nearby, and be ready to provide the information listed above.

Sending a Fax

Fax the information listed above with your name, telephone number, and address to the appropriate Data I/O Customer Support fax number listed at the front of the Preface.

Sending E-mail

To reach Data I/O via e-mail, send a message including your name, telephone number, e-mail address, and the information listed above to one of the following addresses:

```
techhelp@Data-IO.COM
```

or

```
{apple|decwrl|rutgers|gatech|uunet}!pilchuck!techhelp
```

Note: Select one of the five addresses listed above in braces. For example, you might send e-mail to the following address: `uunet!pilchuck!techhelp`.

See your system administrator if you need more information on which address to use.

Using the BBS

To reach Data I/O via the BBS, include your name, telephone number, e-mail address, and the information listed above in a message, and send it to the BBS as described in the following section.

Bulletin Board Service

From the Data I/O Bulletin Board System (BBS) you can obtain a wide range of information on Data I/O products, including current product descriptions, new revision information, technical support information, application notes, and other miscellaneous information.

Using the BBS, you can access device support information, request support for a particular device, and leave messages for the BBS system operator, Customer Support personnel, or other customers. The BBS also includes many downloadable DOS utilities.

Multiple lines are available, all supporting 1200/2400/9600/19200 baud, with U.S. Robotics Dual/HST V.32*bis*/V.42*bis* modems. The modems are set to 8 data bits, 1 stop bit, and no parity. Online help files provide more information about the BBS and its capabilities.

BBS numbers for all countries are as follows:

Country	BBS Number
Belgium	+32-(0)1-6380731
Canada	+1-905-678-0547
France	+33-(0)13-9562699
Germany	+49-(0)89-8585833
Japan	+81-33432-6996
Netherlands	+31-(0)40-58242455
Norway	+47-(0)66-780445
Sweden	+46-(0)8-73910377
Switzerland	+41-(0)1-313-0658
United Kingdom	+44-(0)734-448862
United States	+1-206-882-3211

Warranty Information

Data I/O Corporation warrants this product against defects in materials and workmanship at the time of delivery and thereafter for a period of one (1) year.

The foregoing warranty and the manufacturers' warranties, if any, are in lieu of all other warranties, expressed, implied or arising under law, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

Data I/O maintains customer service offices throughout the world, each staffed with factory-trained technicians to provide prompt, quality service. For warranty service, contact Data I/O Customer Support at the numbers listed at the front of the Preface.

Keep Current Subscription Service

Data I/O offers a one-year renewable subscription to keep your product and documentation up-to-date with the latest features and device support. This subscription, called Keep Current subscription service, also incorporates manufacturer-recommended changes to existing device support to maintain optimum yields, throughput, and long-term reliability.

For more information, or to order Keep Current subscription service, contact Data I/O Customer Support at the numbers listed at the front of the Preface.

Repair Service

After the warranty period expires, repair services are available at Data I/O Service Centers worldwide. Single instance repairs and fixed price annual agreements that cover all parts and labor needed to correct normal malfunctions are also available. The annual agreements include semiannual performance certification.

For more information, or to order a Repair Service Agreement, contact Data I/O Customer Support at the numbers listed at the front of the Preface.

End User Registration and Address Change

If the end user for this product or your address has changed since the Registration Card was mailed, please notify Data I/O Customer Support at the numbers listed at the front of the Preface. This ensures that you receive information about product enhancements. Be sure to include the product serial number, if available.

Typographic Conventions

Throughout this manual different typographic conventions represent different cases of input and output.

Keys

Keys appear in boxes (for example, [Q]) or as bolded text.

The **Enter** key (or on some keyboards, the **Return** key) is represented by this symbol: [↵].

Key Combinations

An instruction for pressing two keys at once, such as ^Z (Control and Z), is represented by two key boxes separated by a plus, such as [Ctrl] + [Z].

A key combination like [Esc] [Ctrl] + [T] means press and release [Esc], then press [Ctrl] and [T] at the same time.

Variable Input

Variable input is italicized and should be replaced with the requested information. For example, "enter **copy** ***filename.hex***" means type **copy** just as you see it and replace ***filename.hex*** with the name of your file.

Optional Input

Optional items of a command are shown in brackets.

[*option1*] [*option2*]...[*optionn*]

Items separated by a vertical bar (for example, OR | OR | ...) are mutually exclusive; that is, only one of the options listed can be specified.

Displayed Text

Text displayed on an LCD or screen appears in a typewriter-like typeface.

```
You will see this text displayed on the screen.
```

Specifications

Note: Unless otherwise noted, the specifications listed below apply to both the 2900 Programming System and the 3900 Programming System.

Functional

RAM	2 MB standard (8 MB optional)
Disk Format	Quad-density, dual-sided 3.5-inch disk with 135 tracks per inch. 1.44 MB formatted. (80MB available with the optional Mass Storage Module)
Controller	Motorola 68000 16-bit microprocessor
Terminal Support	Interfaces with ANSI 3.64 compatible terminals, IBM PCs and compatibles running a terminal emulator program, and many popular ASCII terminals.
Communication Standard	RS-232C
SmartPort	Automatic DTE/DCE port configuration
Data Transfer Rate	110 to 19.2 K baud (up to 115.2 K baud using HiTerm)

Power Requirements

Operating Voltages	100V ac to 240V ac ±10%
Frequency Range	48 to 63 Hz
Power Consumption	200 VA maximum 120 Watts maximum
Input Current	2A maximum

Physical and Environmental

Dimensions	9.53h x 28.58w x 41.28d cm 3.75h x 11.25w x 16.25d inches	
Weight	2900:	Operating: 3.86 kg (8.5 lb) Shipping: 6.14 kg (13.5 lb)
	3900:	Operating: 5.27 kg (11.6 lb) Shipping: 7.53 kg (16.6 lb)
Temperature	Operating:	+4°C to +40°C (+40°F to +105°F)
	Storage:	+4°C to +50°C (+40°F to + 122°F)
	Transportation:	-40°C to +55°C (-40°F to +130°F)
Relative Humidity	Operating:	20 to 80% noncondensing
	Storage:	10 to 90% noncondensing
Altitude	Operating:	To 5,000 meters
	Storage:	To 15,000 meters

Electrostatic Discharge (ESD)

IEC 801-2 (± 15 kV)

Safety

The 2900 and 3900 are both certified by UL, CSA, and TUV to comply with the following safety standards:

Underwriters Laboratories — UL 1950

Canadian Standards Association — CSA C22.2 No. 231

International Electrotechnical Commission — IEC 348 and IEC 1010-1

Verband Deutscher Electrotechniker — VDE 0411

Electromagnetic Emissions

The 2900 and 3900 are both certified to meet VDE 0871 Limit B (vfg 1046).

The 2900 and 3900 are both certified to meet Federal Communications Commission (FCC) Part 15, Class B.

This equipment has been tested and found to comply with limits for a Class B digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instruction manual may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Do not modify the programmer or operate without properly shielded I/O cables (such as the cable supplied with your equipment). Modifying the programmer in any way not expressly authorized by Data I/O or using it without a proper I/O cable could void your authority to operate this equipment under FCC compliance.

Note: FCC Part 15 requires that all "Industrial and Commercial" equipment, such as the 2900 and the 3900, meet the more relaxed Class A limits. However, both the 2900 and the 3900 meet more stringent Class B limits.

Calibration

The 2900/3900 verifies internal voltages every time it is powered up and every time a complete self-test is run. The voltage verification is performed by software and is compared to a laser-trimmed voltage reference. Data I/O recommends that you cycle power AND run a complete self-test cycle at least every three months. See the "Checking Calibration" section of Chapter 5 for more information on checking the reference voltages and the master clock.

To ensure that your 2900/3900 continues to meet product performance specifications, Data I/O recommends that your programmer be returned to an authorized Data I/O Service Center every twelve months for a complete performance evaluation.

3. Insert the Boot Disk

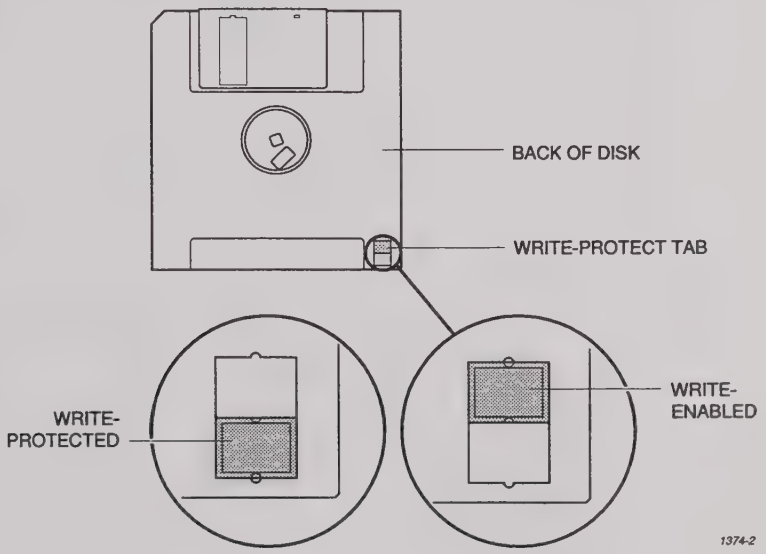
Follow the steps below to insert the Boot disk into the programmer.

1. The 2900/3900 software package contains the 2900/3900 disks. The first time you open the software package do the following:
   a. Remove the 2900/3900 disks from the software envelope.
   b. Fill out and return the End User Registration Card located in this manual. After registering with Data I/O, you will receive information about all future updates and product upgrades.

Note: Please fill out and return the End User Registration Card so we can keep you informed of product updates.

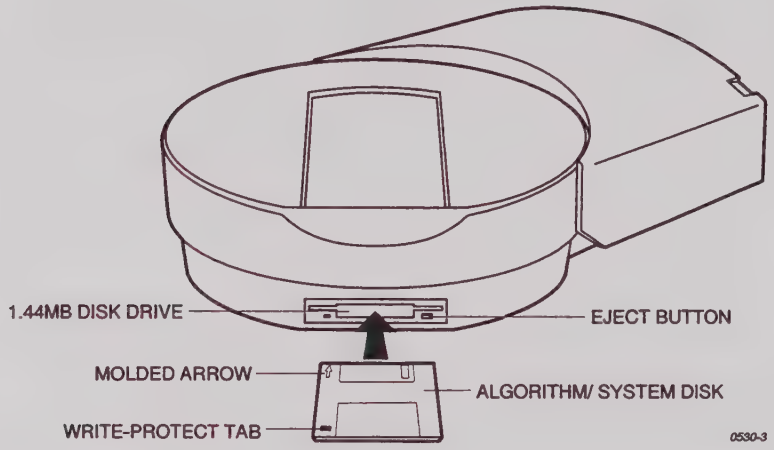
2. Make sure both the Boot disk and the Algorithm/System disk are write enabled. Figure 2-4 illustrates how to write enable a disk.

Figure 2-4
Write Enabling a 3.5-inch Disk



3. As shown in Figure 2-5 insert the Boot disk into the drive, making sure the arrow molded into the plastic case is on the top of the disk and is pointing toward the programmer. Push the disk straight into the drive until the disk drops down and the eject button pops out.

Figure 2-5
Inserting the Boot Disk



About the Boot Disk

The Boot disk contains the system software and the configuration files required by the system software when you boot the programmer. Once the programmer has finished booting, you will be prompted to remove the Boot disk and insert the Algorithm/System disk.

About the Algorithm/System Disk

The Algorithm/System disk contains the system software and all the programming algorithms for the devices currently supported by the programmer. Unless you have the RAM Device Selection feature enabled, the Algorithm/System disk must be installed each time you select a device.

Note: Do not attempt to use the Boot disk or the Algorithm/System disk with more than one 2900 or 3900. Each disk is configured to work only with one particular programmer. Once you get the programmer set up, make a backup copy of the Boot disk and the Algorithm/System disk.

Powering Up

Power up the programmer. The power switch is located on the back panel. When you turn the power switch on, the Power LED should light. If it doesn't, turn the programmer off, check the power connections, and turn the programmer on again.

Note: Do not remove the **Boot** *disk while either the Self Test or disk drive LED is lit. If you remove the Boot disk during powerup, you will need to reboot the programmer.*

Did the Programmer Pass Self-test?

The programmer always performs a self-test upon powerup. The four front panel LEDs illuminate in different patterns, depending on the results of the self-test. The programmer has completed powerup when the Self Test LED and disk drive LED are off. If the Self Test LED does go off, then go to the section titled "Are the Right LEDs Lit?"

If the powerup self-test detected anything wrong, the four LEDs illuminate in different patterns, telling you what the self-test found. In general, if one or more front panel indicators is blinking after the self-test, there may be a faulty circuit board in the programmer. Contact your nearest Data I/O Customer Support office for more information. The different combinations of blinking LEDs are explained below.

Indicator				
Power	**Terminal**	**Remote**	**Self-Test**	**Description**
On	Off	Off	On	Self-test in progress.
Off	X	X	On	Bad power supply
On	Blinking	Blinking	Blinking	Power Supply problem; check voltage selector.
On	Blinking	Off	On	Bad CPU or EPROM
On	Off	Blinking	On	Bad system RAM
On	Blinking	Blinking	On	Bad serial port DUART
X	X	X	Blinking	Bad LCA
On	On	Off	Off	Terminal port OK; self-test finished. No errors.
On	Off	On	Off	Remote port OK; self-test finished. No errors.
On	On	On	Off	Remote & terminal ports OK; self-test finished. No error condition.

Note: X = don't care condition.

"Terminal port OK" means that the programmer has detected an RS-232 serial device connected to the Terminal port and that a valid connection has been established with that device.

Are the Right LEDs Lit?

If the programmer completed the self-test successfully, the Power LED will be lit. Also, if you have equipment connected to the Terminal and Remote ports, the corresponding LEDs should be lit. If all the right LEDs are lit, go to the section titled "Is the Power-on Screen Displayed?"

If the Remote LED and/or Terminal LED should be lit and they are not, check the connections between the programmer and the connected equipment.

Checking the Connections

Sometimes problems are caused by unconnected cables. Turn the programmer off and check all of the following:

- Power cords – Are they all plugged into a live outlet and into the equipment?
- Cables – Is each cable between the programmer and a peripheral connected properly? Is each cable connected to the proper port?
- Terminal – Is the terminal plugged in and turned on? Are the display controls adjusted to allow viewing? Is the terminal an approved terminal type? (See the list of approved terminals for more information) Are the communication parameters set correctly? Is the cable connected to the proper port?
- Host – Is it plugged in and turned on? Is the terminal emulation software installed and configured properly? Is the terminal emulation software running properly? Are the display controls on the monitor adjusted to allow viewing?
- Disk – Is the Algorithm/System (for the 2900) or the Boot disk (for the 3900) inserted properly?
- Base – Is there a Base inserted into the programmer? Is it inserted properly? Is the Base empty? (The Base must be empty for the programmer to boot up properly.)

Now, after checking everything described above, reboot the programmer and go to the beginning of this section, "Power Up the Programmer."

Rebooting the Programmer

If you need to reboot the programmer, you can do it in one of two ways. Either turn the programmer off, wait a few seconds, and then turn the programmer on again. Or, press Esc Ctrl + W . The programmer always performs a powerup self-test whether you do a cold boot (cycling power) or a warm boot (pressing Esc Ctrl + W).

Is the Power-on Screen Displayed?

If the Power-on screen (shown in Figure 2-15) is displayed, you have successfully booted up the programmer. Go to the next step, "6. The Power-on Screen."

If you do not see the Power-on screen, or if you see random characters, the programmer is not communicating properly with your controlling terminal/workstation. One possible cause of the random characters is that the baud rates of the programmer and the controlling equipment do not match. In this case, follow the procedure in the next section to execute the AutoBaud function to "sync up" the baud rates.

Note: If the screen is blank or if random characters appear after you press Enter, the Terminal type configuration is incorrect. Switch the power off and then power up the programmer. Then return to the beginning of this section and select the proper terminal type.

If you do not see your terminal listed on the screen and do not know what terminal type(s) it can emulate, refer to the Compatible Terminals list shown previously in this chapter.

After you have changed the terminal type for this current session, the programmer responds with the following prompt:

```
Save terminal type as power on default? (Y/N) [N]
```

3. Press [N][↵] to go to the Main Menu without changing the default terminal type.

When the Main Menu appears, the programmer is ready for operation. Go to the section titled "Insert Algorithm/System Disk."

Changing the Default Terminal

Follow the steps below to change the default terminal type.

1. At the bottom of the Power-on screen, the programmer displays the following prompt:

   ```
   Do you want to select a new terminal type? (Y/N) [N]:
   ```

   Press [Y][↵] to select a new terminal type.

2. The programmer then displays the default and current terminal types and a list of the available terminal types. To select a new terminal type, enter the number corresponding to that terminal type and press [↵].

Note: If the screen is blank or if random characters appear after you press Enter, the Terminal type configuration is incorrect. Switch the power off and then power up the programmer. Then return to the beginning of this section and select the proper terminal type.

If you do not see your terminal listed on the screen and do not know what terminal type(s) it can emulate, refer to the Compatible Terminals list shown previously in this chapter.

After you have changed the terminal type for this current session, the programmer will respond with the following prompt:

```
Save terminal type as power on default? (Y/N) [N]
```

3. Press [Y][↵] to change the default terminal type. After the programmer saves the currently selected terminal type as the power-on default, the Main Menu appears, indicating that the programmer is ready for operation. Go to the section titled "Insert Algorithm/System Disk."

Insert Algorithm/System Disk

Once your programmer has finished booting (the self test is finished and the drive LED is off), remove the Boot disk and insert the Algorithm/ System disk, which contains all the programming algorithms for the devices currently supported by the 2900/3900. This disk must be in the disk drive when you select a device.

Note: You do not have to have the Algorithm/System disk in the disk drive if you have the Mass Storage Module installed in your 2900/3900.

Once you have inserted the Algorithm/System Disk, go to the next step, "Setting Up High Speed Download."

Setting Up High Speed Download

This section explains how to set up the programmer so you can download files to the programmer at 115.2K baud.

To take advantage of the High Speed Download feature, you must be controlling the programmer from a PC, the PC must be connected to the programmer's Remote port, and you must be using HiTerm as your terminal emulation software. See the section titled "Installing HiTerm" earlier in this chapter.

Skip to the section titled "What to Do Next Time" if you are not using HiTerm on a PC.

Why Should I Use High Speed Download?

High Speed Download can reduce the transfer time for a large data file by as much as 92%.

As an example, downloading a formatted data file for a 1MB EPROM takes 6 minutes and 38 seconds at 9600 baud. With High Speed Download, it only takes 32 seconds to download the same data file from a 25 MHz 386-based PC.

Note: To obtain the best results, we suggest you run HiTerm on at least a 286-based PC.

What Happens During High Speed Download?

When you begin a download to the programmer, the programmer and HiTerm communicate with each other to coordinate the transfer. HiTerm translates the data file on the PC into a special binary format. The compression of the data file reduces the size of the data file by up to 64%.

Note: HiTerm compresses a temporary copy of the data file; your original data file is not changed.

HiTerm then switches the baud rates on the PC and the programmer to 115.2K baud and downloads the data file to the programmer. After the download, the baud rates on the PC and programmer are restored to their original values.

What to Do Next Time

Next time you power up the programmer, you probably do not need to follow all the steps outlined in this chapter. Listed below are the normal steps for preparing for another session on the programmer.

Note: If you have not used the programmer for awhile, or if you suspect the programmer might have been moved from one area to another, follow the procedure below before you use the programmer.

To prepare the programmer for another session, follow the procedure below:

1. Check the power cords and cables between the programmer and the connected equipment.
2. If you are controlling the programmer from a PC or workstation, make sure it is on and that the terminal emulation software (such as HiTerm) is running.

   If you are controlling the programmer from a terminal, make sure it is on.

   If you are using PROMlink-6, select **VT100 on Programmer Port** from the **Utilities** menu.
3. Insert the Boot disk into the disk drive.
4. Select and insert a Base into the programmer. Make sure the Base is locked in place.
5. Power up the programmer.
6. Verify the terminal type when the Power-on screen appears.
7. When prompted, remove the Boot disk and insert the Algorithm/System disk.

You are now ready to begin a new session on the programmer.

Note: If you are using the 3900 and are saving power-up defaults, you will be prompted to insert the Boot disk.

3. Next, enter a description of the configuration file you are saving, for example, **Config file for Intel 27C256**. The name can be up to 30 characters long. After you enter the file description, press ↵ . The programmer displays the following message in the message bar:

   `Parameter Entered`

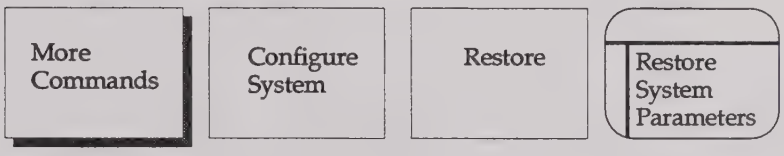
4. Finally, to save the current system configuration to a configuration file, press ↵ . While saving the configuration, the action symbol rotates. When done, the programmer displays the following message:

   `System parameters saved`

Note: If a device has been selected, it is saved as part of the configuration file.

Restore System Parameters

System Parameters are all the parameters on the Programming, Serial I/O, Communication, and Interface screens.



With the Restore command, you can restore a set of previously saved system parameters.

To restore a system configuration, follow these steps:

1. Select Restore from the Configure System Parameters menu.
2. The programmer displays a list of all the configuration files that have been saved on the Algorithm/System disk. Look at the list of configuration files and find the file number of the file you want to restore.
3. Enter the file number of the configuration file you want to restore and press ↵ . The programmer loads the system parameters and displays the following message in the message bar when done:

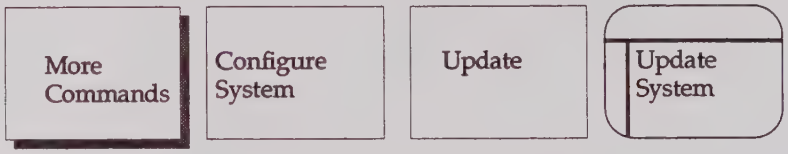
   `System parameters restored`

Note: When you restore a configuration, the device algorithm selected when the configuration file was saved (if any) will also be restored.

When you return to a screen, the cursor will be where you left it. Restoring parameters to factory defaults or power-up defaults returns the cursor to its original position. Restoring parameters to user-defined defaults has no effect on cursor positions.

Update

Use this command to update your programmer to a new version of system software.



When Should I Use This Command?

You should use the Update command when you have a new version of system software you want to use with your programmer.

In addition, you can also use the Update command to install the system software on the Mass Storage Module (if one is installed in your programmer) and configure your programmer to boot from the MSM.

Note: When you first receive your programmer, you do not need to run the Update command, because the system software is already installed.

Before You Begin

1. Make sure a Base is installed in the programmer.
2. Make sure the cable connecting the programmer and the controlling terminal/PC/workstation is connected to the **Terminal** port on the programmer.
3. Make sure that the communication parameters on your controlling terminal/PC/workstation are set as follows:
   - 9600 baud
   - 1 stop bit
   - 8 data bits
   - No parity

   Later, when you are finished installing the new version of system software, you can reconfigure your programmer as you had it before the update.
4. Insert the NEW Boot disk in the programmer's disk drive. Make sure the disk is write enabled.
5. Power up the programmer. If the programmer is already powered up, reboot it by pressing Esc Ctrl + W or by cycling power.

Updating the System ROM

CAUTION: Turning off the system power or removing the system disk while the programmer is updating system ROM damages the programmer.

6. When the programmer reboots, the following message appears:

```
Disk revision does not match ROM
        Disk revision = y.yy
        ROM revision = x.xx

Do you want to update ROM to new disk revision?(Y/N)
```

Press [Y] to update the system ROM. The programmer responds with the following message:

```
Current ROM contents will be over-written!!

WARNING:  Removing power during update process
            damages the X900!

Are you really sure? (Y/N)
```

Press [Y] to confirm the update of the system software. The programmer updates the system ROM to reflect the new version of system software.

7. If you answer [N] to the above question, the programmer displays the following message:

```
Insert correct disk revision.

Press Esc Ctrl-W to reboot.
```

If you still want to install new software, return to step 4. Remember, you must install the new system software before you can use it.

8. If you answered [Y] to both questions shown in step 6, the programmer goes through the following steps:

- Loading System Data
- Erasing System ROM
- Programming First Half
- Programming Second Half
- Verifying Data
- Rebooting

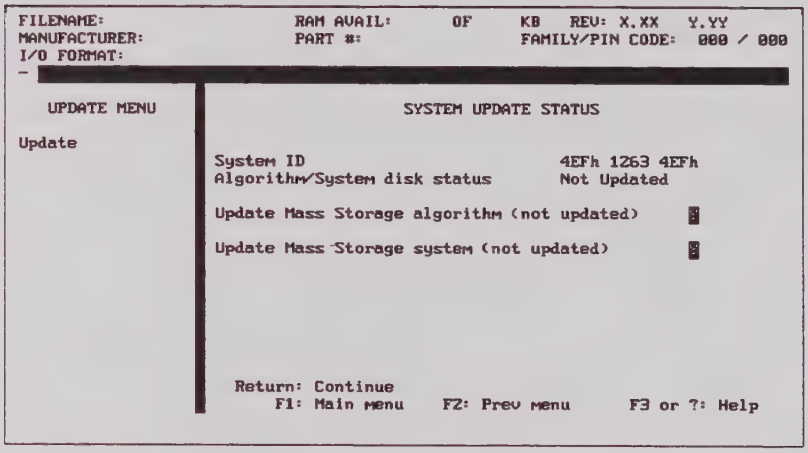
The programmer displays the current step on the screen. When finished, the programmer reboots.

9. After booting up, the programmer displays the banner screen and prompts you to select a terminal type that matches your configuration.

10. If prompted, remove the Boot disk and insert the NEW Algorithm/System disk.

11. After selecting a terminal type the System Update screen appears, as shown in Figure 5-2.

Figure 5-2
The System Update Screen



Note: The two `Update Mass Storage` *lines shown in the above illustration appear only if you have a Mass Storage Module installed in your programmer.*

12. If you have a Mass Storage Module installed in your programmer, the programmer defaults to installing the system and algorithm files on the Mass Storage Module. If you DO NOT want to install the system and algorithm files on the Mass Storage Module:

    - Move the cursor to the `Update Mass Storage system` prompt and press [N].
    - Move the cursor to the `Update Mass Storage algorithm` prompt and press [N].

13. Press ↵ to install the new system software. The action symbol rotates while the programmer is working. If prompted, remove the Algorithm/System disk and insert the NEW Boot disk.

    If you are installing the system files on the MSM, you will be prompted to remove the Algorithm/System disk and insert the Boot disk.

CAUTION: ***Do not remove disks from the programmer during the update operation unless the programmer prompts you to do so.***

14. When the programmer is done, `OPERATION COMPLETE` is displayed in the message bar.

Note: It takes about 5 minutes to install the algorithm and system files on the Mass Storage Module. If you are not installing the algorithm and system files on the Mass Storage Module, the update procedure takes about 30 seconds.

Verifying the Update

Follow the steps below to verify that you installed the software update properly.

1. If the `Algorithm/System disk status` field reads `Updated`, the update was installed properly.

   If the `Algorithm/System disk status` field reads `Not Updated`, the update was not installed properly. Contact Data I/O Customer Support if the update did not install properly.

2. If you installed the new algorithm files on the Mass Storage Module, the `Update Mass Storage algorithm` prompt should now contain the version number of the new algorithm files in parenthesis.

3. If you installed the new system files on the Mass Storage Module, the `Update Mass Storage system` prompt should now contain the version number of the new system software in parenthesis.

You are finished with the installation of the new version of system software.

PROMlink-6 and TaskLink Users

If you are using PROMlink-6 or TaskLink, remember to select the **Update Device List** command from the **Utilities** menu in PROMlink-6 or TaskLink after updating your programmer to a new version of system software.

When You Are Finished

When you are finished updating system software, we suggest you make a back-up copy of your new system disks. See the Duplicate Disk command for more information.

If you have access to a DOS-based PC with a 1.44MB disk drive, we suggest that you use the DOS DISKCOPY command to make a copy of your new disks. If you use DOS, make sure you use DISKCOPY and not COPY. The backup must be an exact, bit-for-bit, sector-for-sector copy of the original. Store the backup copy in a safe place.

Note: If you use DISKCOPY, the destination disk must be formatted on the programmer before copying. See the description of the Duplicate Disk command for more information.

Also, you should check your job files to make sure they produce the same results as they did before you updated your system. We suggest that you re-record your job files when you change versions of system software or when you add libraries. For more information see the description of the Job Files command later in this chapter.

If you changed your system configuration before you proceeded with the update, you should return your system to its prior configuration.

The section titled "Carrying a Configuration File Forward" describes how to use your system configuration files with the new version of system software you just installed.

Carrying a Configuration File Forward

Beginning with version 2.20 (2900) or version 1.60 (3900) you can carry system configurations forward after you update to a new version of system software. Previously, you had to recreate system configurations when you updated to a new version of system software.

Note: If you carry the system configurations forward, the powerup defaults will be carried forward in addition to any other user-defined system configurations. This is because the powerup defaults, in addition to the factory defaults and any user-defined configurations, are stored in a file named ***sysparm.sys*** *on the Algorithm/System disk.*

Now, the following scenario is possible:

- Under version X.3, create and save two different system configurations, such as one for memory devices and one for logic devices.
- Update to version X.4 system software.
- Carry the system configurations created under version X.3 forward to X.4.
- Use the two system configurations with version X.4.

Note: This procedure works only with system software version 2.20 (2900) or version 1.60 (3900) and later.

For more information on saving and restoring system configurations, and on system configurations in general, see the "Restore System Parameters" and the "Save System Parameters" sections in this chapter.

What You Need

To carry system parameter settings forward, you will need a DOS-based IBM-compatible PC with a 3.5-inch floppy drive capable of reading and writing 1.44 MB floppy disks.

Carrying the Configuration File Forward

Follow the steps below to carry your system configurations forward to a new version of system software.

Note: In the following steps "old" refers to the Algorithm/System disk containing the system configurations you want to carry forward to the "new" Algorithm/System disk.

1. If you are updating to a new version of system software, install the update first. See the "Update" section of this chapter for more information.
2. Make a backup copy of the new Algorithm/System disks. See the description of the Duplicate Disk command for more information on making a backup copy of your Algorithm/System disks.

Note: Store the master copies in a safe place. Use your backup copies for day-to-day operation.

3. Power up the PC. When the PC has finished booting, insert your working copy of the old Algorithm/System disk in the 3.5-inch drive in the PC.

Note: Make sure you insert the Algorithm/System disk with the configurations you want to carry forward.

4. Copy the sysparm.sys file to your hard drive. Enter the command

   copy *drive1*:**\sysparm.sys** *drive2*:

   where *drive1*: is the drive designator for the 3.5-inch disk drive and *drive2*: is the drive on which you will be temporarily storing the copy of the sysparm.sys file.

5. Remove the old Algorithm/System disk and insert the new Algorithm/System disk in the 3.5-inch drive. Make sure the new Algorithm/System disk is write enabled.

Note: We suggest you work with the backup copies of your Algorithm/System disks and not with the originals.

6. Copy the sysparm.sys file to the new Algorithm/System disk. Enter the command

   copy *drive2*:**sysparm.sys** *drive1*:**\sysparm.sys**

   where *drive2*: is the drive you copied the sysparm.sys file to in step 4.

   If you are using a 3900, remove the Algorithm/System disk, insert the Boot disk, and copy the **sysparm.sys** file to the Boot disk.

7. If you have more programmers that you want to copy a configuration file to, repeat steps 5 to 6 for each of the remaining programmers. For example, if you have three 3900s in a lab, and if you used the same configuration files on all three, the following scenario would be possible:

   - Update all three 3900s to a new version of system software.
   - Make backups of the new Algorithm/System disks.
   - Perform steps 1 to 6 as described above on one 3900.
   - Repeat steps 5 to 6 for the remaining two 3900s.

8. When finished, remove the new Algorithm/System disks from the 3.5-inch drive and insert it into the programmer. When you use the programmer, the power-up defaults and user-defined configurations you saved will be available for use.

Duplicate Disk

Use this command to duplicate an entire disk.

To duplicate an entire disk, you can use either the Duplicate Disk command, or you can use the DOS DISKCOPY command.

Using DOS

If you have access to a DOS-based PC with a 1.44MB 3.5" disk drive, we suggest that you use the DOS DISKCOPY command to make a copy of your Algorithm/System disk. If you use DOS, make sure you use DISKCOPY and not COPY. The backup must be an exact, bit-for-bit, sector-for-sector copy of the original. For more information, see your DOS manual.

Note: If you use the DOS DISKCOPY command to duplicate a disk, make sure that the destination disk has been formatted on the programmer before the copy is made. See the More Commands/File Operations/Format Disk command for more information.

Using the Programmer

To use the programmer to duplicate a disk, follow these steps:

Note: Whether you use DISKCOPY or Duplicate Disk, we recommend you write protect your source disk by sliding the write protect tab on the disk so you can see through the hole which it exposes. See your DOS user's manual for more information on duplicating a 1.44MB disk.

1. Insert the disk you want to duplicate into the disk drive.

CAUTION: Do NOT use the Algorithm/System disk or the Boot disk as the destination disk, because the original contents of the destination disk will be lost.

2. If you want to duplicate the disk, move the cursor to the `Are You Sure?` field and press [Y].

Note: The source disk and destination disk parameters are currently fixed at A.

CAUTION: Duplicating a disk erases the contents of the destination disk. Also, this operation uses RAM as a temporary storage buffer and alters the contents of RAM.

3. Move the cursor to the `Do you want to verify disk?` field and enable the disk verification procedure.

   To disable the verification, and speed up the process, press [N]. To enable the verification, press [Y].

Note: Although disk duplication takes longer if the verification option is turned on, we suggest that you turn verification on while duplicating an important disk, such as the Algorithm/System disk or the Boot disk.

4. Finally, press [↵] to begin the disk duplication. During the duplication you are prompted when to swap disks.

Note: After every disk duplication operation, the verification parameter returns to **Y** *and the* `Are you sure` *parameter returns to* **N**.

Format Disk

Use this command to prepare a data disk for use. A disk must be formatted before it can be used.

To format a disk, follow these steps:

1. Insert the disk to be formatted into the disk drive.
2. When you are ready to format the disk, type [Y] [↵] at the `Are You Sure?` prompt.

   If you are not ready to format a disk, press [F2] to return to the File Operations menu.

3. The programmer checks the disk in the disk drive, making sure it is not a System disk. If the programmer detects the Algorithm/System disk or the Boot disk, this message appears.

```
WARNING: system disk in drive. Hit return to continue,
^Z to abort.
```

CAUTION: Do NOT format the programmer Algorithm/System disk or the Boot disk, because the original contents of the disk will be lost during formatting.

To format the disk, press [↵]. If you do not want to format the disk, press [Ctrl] + [Z].

Summary of Standard CRC Commands

Command	Description	Response
–	Invert RAM	>
hhhhhh:	Select device begin address	>
hhhhhh;	Select memory block size	>
hhhhhh<	Select memory begin address	>
nn=	Select I/O timeout	>
fffppp@ or *ffpp@*	Select device type	>
*cff*A	Enter translation format	>
B	Blank check	>
C	Compare to port	>
D	Set odd parity	>
E	Set even parity	>
F	Error status inquiry	HHHHHHHHH>
G	Configuration inquiry	DD>
H	No operation	>
I	Input from port	>
J	Set 1 stop bit	>
K	Set 2 stop bits	>
L	Load RAM from device	>
*hh*M	Enter record size	>
N	Set no parity	>
O	Output to port	>
P	Program device	>
Q	Swap nibbles	>
R	Return status of device	AAAAA/BB/C>
S	View sumcheck	HHHH>
T	Illegal-bit test	>
*hh*U	Set nulls	>
V	Verify device	>
*hhhhhhhh*W	Set I/O offset	>
X	Error code inquiry	HH....HH>
Y	Display parity errors	HHHH>
Z	Exit remote control	none
[	View device family/pinout code	FFFPPP>
\	Move memory block	>
hh^	Clear/fill RAM with data	>

Summary of Extended CRC Commands

Command	Description	Response
01]	Display system configuration	SSSS/AAAA/MM/PP/II/JJ>
*nn*02]	Set upload wait time	>
*n*03]	Set device ID verify option	HHHHHHHH> or >
*nn*04]	Set Remote port baud rate	>
*xxx...xxxx*05]	Set host command	>
*n*06]	Select data bits	>
*n*07]	Set next set member	>
*nn*22]	Set data word width	>
*n*23]	Select number of verify passes	>
*n*24]	Select security fuse programming option	>
*n*26]	Specify logic verify options	>
*n*27]	Set/clear enable/disable sec. fuse	>
*n*28]	Fill fuse map	>
*n*29]	Set reject count option	>
*hhh*2A] or *hh*2A]	Enable programming options	>
*hhh*2B] or *hh*2B]	Disable programming options	>
*nhh*2C]	Select memory fill option	>
*hh*2D]	Vector test options	>
2F]	Return 8-character sumcheck	HHHHHHHH>
*xxx...xxxx*30]	Set data file name	>
*n*31]	Set data source/destination	>
*xxx...xxxx*33]	Select device manufacturer	>
*xxx...xxxx*34]	Select device part number	>
*xxx...xxxx*38]	Load file from disk	>
*xxx...xxxx*3B]	Delete disk file	>
*n*3C]	Set data transfer port	>
*xxx...xxxx*3E]	Select Keep Current algorithm	>
*n*40]	Upload device information	See text following
*n*41]	Upload self-test results	AAA...AA>
43]	Upload yield tally	See text following
46]	Clear yield tally	>
49]	Suspend CRC mode	Displays terminal screen
*n*4A]	Get filename from disk	AAA...AA>
*n*4D]	Select algorithm source	>
*n*52]	Select algorithm source for floppy disk or MSM	>
54]	Upload device footnote	See text following
55]	Upload device-specific message	See text following
56]	Upload memory verify failure	ddPAAAAAAAAHHhh
57]	Get checksum of operation	See text following
58]	Upload system ID	HHHH HHHH HHHH>
A7]	Swap bytes	>
DC]	Device check	See text following
EB]	Input JEDEC data from host	>
EC]	Output JEDEC data to host	>
FC]	Restore CRC entry default parameters	>
FD]	Restore user-defined CRC parameters	>
FE]	Save user-defined CRC parameters	>

Standard CRC Commands

The Standard CRC Commands are described in this section. Note that the lowercase letters preceding the commands are arguments that must be specified according to the options listed under the corresponding command description. Except where noted, the standard CRC commands use the following notation conventions:

- lowercase letters indicate arguments that must be specified
- *h* represents a hexadecimal digit.
- *n* represents a decimal digit.
- *xxx...xxxx* represents a string of characters or a filename. Filenames are limited to 14 total characters in length: drive specification (2 characters) plus name (8 characters) plus extension (3 characters preceeded by a dot).

For example, ***nn*02]** indicates that you may precede the **02]** command with two decimal digits.

— **Invert RAM** — Inverts the data in RAM within the address range defined by the Beginning Memory Address and the Memory Block Size. If the Memory Block Size is set to 0, all of User Memory is inverted.

***hhhhhh*:** **Select Device Begin Address** — Sets the first device address to load, program, or verify. This command is also used as the destination address in a RAM to RAM block move. The Device Begin Address defaults to 0 if no address precedes the colon.

***hhhhhh*;** **Select Memory Block Size** — Sets the number of bytes to be uploaded or programmed. This command sets both the Memory Block Size and the Device Block Size. If no argument precedes the semicolon, or if the value of the argument is 0, the Memory Block Size is set to the size of User Memory and the Device Block Size is set to the size of the currently selected device. If the value of the argument is greater than the physical device size, Device Block Size is set to the size of the entire device.

***hhhhhh*<** **Select Memory Begin Address** — Sets the first RAM address from which or to which data will be transferred. This address is also used as the Begin RAM Address where the programming data is located. The Memory Begin Address defaults to 0 if no address precedes the <.

***nn*=** **Select I/O Timeout** — Specifies the number of seconds the programmer will wait during a download before it returns an I/O timeout error (CRC error code 46). Valid arguments range from 01 to 99 seconds. To disable the I/O Timeout, either specify an I/O Timeout of 00 seconds, or send a null value (i.e., just send the = command). The I/O Timeout defaults to 30 seconds.

fffppp@ or *ffpp*@	**Select Device Type** — Selects the family/pinout code for a device operation. You can specify either 3-digit family/pinout codes, *fffppp*, or 2-digit family/pinout codes, *ffpp*. To select a device by manufacturer and part number, see CRC commands **33]** and **34]**.
*nff*A	**Enter Translation Format** — Selects the instrument control code *n* and the data translation format *ff* to be used for I/O data transfers through the Remote port. If 1 or 2 digits precede the **A**, the digits select the data translation format and the instrument control code defaults to 0. If 3 digits precede the **A**, the first digit designates the instrument control code and the last two digits specify the data translation format. For example, sending **191A** selects instrument control code 1 and data translation format 91. The data translation formats and the instrument control codes are described in Chapter 7.
B	**Blank Check** — Performs a blank check on the currently socketed device.
C	**Compare to Port** — Compare data in the programmer's RAM with data received through the Remote port using the current data translation format. (JEDEC format cannot be used with this command: this command works only for memory formats.) The current Memory Begin Address and I/O Offset are used to calculate the RAM address where the data is located to compare against the incoming data.
D	**Set Odd Parity** — Sets odd parity for serial data transfers through the Remote port.
E	**Set Even Parity** — Sets even parity for serial data transfers through the Remote port.
F	**Error Status Inquiry** — Returns a 32-bit number in the format HHHHHHHH, where each H is a hex character. The 32-bit word defines the accumulated errors in the error status word since the last **F** command. See the section titled "Error Status Word" in this chapter for more information.
G	**Configuration Inquiry** — Returns the configuration information in the form DD, where DD is the disk version. For example if 13 is returned, the disk version is 1.3. Additional configuration information can be obtained by using the extended command **01]**.
H	**No Operation** — Returns the > prompt followed by a ↵ and, if specified, a line feed. No operation is performed.

*n*03]

Select Electronic ID — Enables/disables the electronic ID test. Also, this command can return the Electronic ID of the selected device. The argument must be one of 0, 1, or 2, where

0 Disables electronic ID.

1 Enables electronic ID.

2 Returns electronic ID as eight hex digits in the form hhhhhhhh. Leading zeros are sent for those devices not having an 8-digit ID (0000890D, for example).

*nn*04]

Set Remote Port Baud Rate — Sets the baud rate for the Remote port. Valid arguments are listed and described below:

nn	Baud Rate	*nn*	Baud Rate
01	50	10	1500
02	75	11	1800
03	110	12	2000
04	134.5	13	2400
05	150	14	4800
06	200	15	7200
07	300	16	9600
08	600	17	19.2K
09	1200		

*xxx...xxxx*05]

Set Host Command — Sets the command string to be sent to the host for upload or download of data. Valid arguments can range in length from 0 to 58 characters. If no argument precedes the **05]** command, no host command is sent. The programmer appends a carriage return to the end of the string.

*n*06]

Select Data Bits — Sets the number of data bits for serial data transfers through the Remote port. Valid arguments are listed and described below:

8 Selects 8 data bits.

7 Selects 7 data bits.

*n*07]

Set Next Set Member — Determines data organization during serial set programming. For example, if you are programming 4-bit devices and have set the Data Word Width to 8, setting the Next Set Member (*n*=2) means that the upper 4 bits of each data byte (rather than the lower 4 bits) is used to program the device.

*nn*22]

Set Data Word Width — Specifies, in bits, the width of a data word in the device being programmed. Valid arguments range from 4 to 64.

*n*23]

Select Number of Verify Passes — Selects the number of verify passes and the type(s) of voltage(s) used during a verify operation. Valid arguments are listed and described below:

0 Specifies no verify passes.

1 Performs a single-pass verify with nominal V_{CC}.

2 Performs a two-pass verify, one at the maximum allowed V_{CC}, and one at the minimum allowed V_{CC} value.

*n*24] **Enable Security Fuse** — Enables/disables programming the security fuse(s). Valid arguments are listed and described below:

0 Disables programming the security fuse(s).

1 Enables programming the security fuse(s).

*n*26] **Specify Logic Verify Options** — Selects the type of logic verification to perform during a verify operation. Valid arguments are listed and described below:

0 Performs the fuse verify test followed by a structured vector test.

1 Performs only the fuse verify test.

2 Performs the structured vector test.

Note: The 2900 does not support vector testing for logic devices with more than 44 pins regardless of the verify option setting.

The 3900 does not support vector testing for logic devices with more than 84 pins regardless of the verify option setting.

*n*27] **Set/Clear Enable/Disable Security Fuse** — Enables/disables programming of the security fuse and sets the state of the security fuse. Valid arguments are listed and described below:

0 Disables programming of all security fuses and sets all security fuse states in RAM to 0.

1 Disables programming of all security fuses and sets all security fuse states in RAM to 1.

2 Enables programming of all security fuses and sets all security fuse states in RAM to 0.

3 Enables programming of all security fuses and sets all security fuse states in RAM to 1.

4 Sets first security fuse state in RAM to 1.

5 Sets second security fuse state in RAM to 1.

6 Sets third security fuse state in RAM to 1.

*n*28] **Fill Fuse Map** — Specifies the fuse state with which to fill the fuse map. Valid arguments are listed and described below:

0 Fills the fuse map in RAM with 0s.

1 Fills the fuse map in RAM with 1s.

*n*29] **Set Reject Count Option** — Selects the maximum number of programming pulses required to program a device before the programmer rejects the device as unprogrammable. Valid arguments are listed and described below:

0 Selects the number of programming pulses specified by the device manufacturer.

1 Selects a single programming pulse or military reject count.

***n*4D]**

Select Algorithm Source — Selects which set of algorithms to use with the *xxx...xxxx*33], *xxx...xxxx*34], *n*40], and @ commands. Valid arguments are listed and described below:

0 Use the set of algorithms included on the Algorithm/System disk.

1 Use the extended set of algorithms (if any are available). Devices are selected from the ALG.EXT file.

The ***n*4D]** command allows you to switch between the algorithms that are included on the Algorithm disk and, for example, a set of extended algorithms. Consider the following scenario:

- Enter CRC mode
- Select the standard device file (*0*4D] command)
- Select a device manufacturer (*xxx...xxxx*33] command)
- Select a device part number (*xxx...xxxx*34] command)
- Program the device (P command)
- Select the Extended algorithm file (*1*4D] command)
- Select a device manufacturer (*xxx...xxxx*33] command)
- Select a device part number (*xxx...xxxx*34] command)
- Program the device with the Extended algorithm (P command)

The above example is a typical example of how to use the ***n*4D]** command to select a different algorithm source.

n4F]

RAM Device Selection — Enables/disables RAM device selection. Valid arguments are listed and described below:

0 Disable RAM device selection

1 Enable RAM device selection; allowed only if the programmer has at least 4 MB of memory.

The 14F] command instructs the programmer to select the device algorithm from a special area 1 MB in size located at the high end of user RAM. After the algorithm file has been loaded into user RAM, the amount of user RAM available to the user is reduced by 1 MB. When the RAM device selection switch is turned on after the programmer is booted, the algorithm file will not be loaded into user RAM until a device is selected.

When a device is selected, if the algorithm file has not been loaded into this special user RAM area, it will be loaded before the device algorithm is selected. This causes the device selection time to be longer than usual for the first device after the switch is turned on. Subsequent device selection operations will take approximately one second, because no more disk accesses are necessary.

---

*Note: PPI Base information file (***adapters.sys***) is not loaded into RAM even if the RAM device selection switch is on. PPI Base information must be accessed from the disk when devices supported on the PPI Base are selected.*

*n*52]

Select Algorithm Source — Depending on the argument supplied, the *n*52] command sets the source for device programming algorithms to either the floppy disk or the Mass Storage Module. Valid arguments are listed and described below:

0 Select algorithms from the floppy drive

1 Select algorithms from the Mass Storage Module

54]

Upload Device Footnote — Results in one of the following actions:

If a selected device has a footnote and **devfnote.sys** is present, the footnote is uploaded. If devfnote.sys is not present, the footnote index list is uploaded. If a selected device has no footnote, error code B6 is returned.

Definition	Number of Bytes
Length of all footnotes to be uploaded in decimal bytes	2
<CR><LF>	2

Next is data for EACH footnote, organized as follows:

Text of footnote	Variable
<CR><LF>	2

If no device is selected, error code 30 is returned.

55]

Upload Device-specific Message — Uploads algorithm dynamic messages and device-specific failure or warning messages. The output format is as follows:

Definition	Number of Bytes
Length of message in decimal bytes	2
<CR><LF>	2

Next is data for the message, organized as follows:

Text of message	Variable
<CR><LF>	2

56]

Upload Memory Failure — The output format for 16-bit devices is as follows:

DDPAAAAAAAAHHHHhhhh

dd	Device socket
P	Verify pass indicator: H = High, N = Nominal, and L = Low V_{CC} Verify
AAAAAAAA	Failed memory location
HHHH	Data in User RAM
hhhh	Data read from the device

The output format for 8-bit devices is as follows:

ddPAAAAAAAAHHhh

HH	Data in User RAM
hh	Data read from the device

This command applies only to memory devices and uploads only the first failed location.

57] **Upload Memory Failure** — 57] returns the checksum of the last successful program, load or verify operation. If the last program, load or verify operation was not successful, the error code BC is returned.

58] **Uploads the System ID to the CRC Port** — The output is in the following format:

HHHH HHHH HHHH

where each *H* is an ASCII hex character. This number is also displayed on the Update screen.

A7] **Swap Bytes** — Swaps the high bytes and the low bytes in a given memory range. Use the *hhhhhh*< and *hhhhhh*; commands to specify the memory begin address and the size of the memory range to swap. The memory begin address (specified by the *hhhhhhh*< command) added to the block size (specified by the *hhhhhh*; command) cannot exceed the size of user memory. Also, the block size must be an even number.

Entering a block size of 0 will swap all memory beginning with the address specified by the *hhhhhhh*< command.

Note: This command will not work if you have a logic device selected.

If you want to swap high- and low-order nibbles, see the **Q** command for more information.

DC] **Device Check** — Checks for the presence of a device in the socket. If the socket is empty, the **DC]** command returns an F. Error code 3B will be returned after the **X** command is sent. If a device is in the socket, further device checks are done if the device supports insertion and socketing tests. For example, the **DC]** command returns a device insertion error if the device is mis-socketed. If a device is in the socket and no continuity errors occur, the **DC]** command returns the normal > prompt.

EB] **Input JEDEC Data From Host** — Sets the I/O Format to 91 and waits for JEDEC-formatted data to be sent to the programmer through the Remote port. This command is valid only for logic devices. All the fuse map and structured vectors will be received and placed in RAM at the corresponding position.

EC] **Output JEDEC Data To Host** — Sets the I/O Format to 91 and sends JEDEC-formatted data from the programmer through the Remote port. This command is valid only for logic devices. The fuse and vector data in RAM must be valid for the selected logic device. The complete fuse map and structured vectors are output.

FC] **Restore CRC Entry Default Parameters** — Sets the CRC parameters to the original factory defaults.

FD] **Restore User-defined CRC Parameters** — Sets the CRC parameters to the last saved user-defined CRC parameters. The user-defined CRC parameters are the same as the CRC factory defaults if no user-defined parameters have been changed.

FE] **Save User-defined CRC Parameters** — Saves the current CRC parameters to a disk as the user-defined CRC parameters.

Error Status Word

The **F** command returns an 8-character, 32-bit error status word. The following table shows the format of the error status word.

The 8-character word is broken into four 2-character groups. The first 2-character group defines receive errors, the second group defines programming errors, the third group defines I/O errors, and the last group is unused. Each 2-character group contains eight bits, with each bit representing an error or error type. For each bit, a 1 represents an error and a 0 represents no error.

	Bit No.	Value	Description
Receive Errors	31	8	ANY ERROR. If the word contains any errors, the most significant bit (bit 31) will be high
	30	4	Not used
	29	2	Not used
	28	1	Not used
	27	8	Not used
	26	4	Serial-overrun error (42)
	25	2	Serial-framing error (41,43)
	24	1	Not used
Programming Errors	23	8	Any device-related errors
	22	4	Not used
	21	2	Not used
	20	1	Not used
	19	8	Device not blank (20)
	18	4	Illegal bit (21)
	17	2	Nonverify (23, 24, 2B, 2C, A2)
	16	1	Incomplete programming (22, 2A, 30, 31)
I/O Errors	15	8	I/O error (46)
	14	4	Not used
	13	2	Not used
	12	1	Compare error (52)
	11	8	Sumcheck error (82)
	10	4	Record-type error (94)
	9	2	Address error (27)
	8	1	I/O Format error (84, 90)
Unused	7	8	Not used
	6	4	Not used
	5	2	Not used
	4	1	Not used
	3	8	Not used
	2	4	Not used
	1	2	Not used
	0	1	Not used

Example:

What errors are indicated in this error status word: 80888000 ?

8 — the word contains error information
0 — no receive errors
8 — device related error
8 — device is not blank (error 20)
8 — I/O error
0 — no errors
0 — no errors
0 — no errors

Note: 1. *The numbers in parentheses are the programmer error codes, defined in the error codes section of this chapter.*
2. *An error can cause as many as 3 bits to be high: the bit which represents the error, the most significant bit of the 8-bit word in which the error bit occurs, and the bit 31.*
3. *After being read, the error-status word resets to zero.*

CRC Error Codes

Following is a list of error codes that appear while the programmer is being operated in the computer remote control mode. These error codes will be returned by the programmer after it receives the **X** command. Normally, you should send the **X** command after the programmer sends an **F** in response to a command. The list is in numerical order, according to the error code.

1F — **Cannot erase device error** — Appears after the programmer was not able to erase an EEPROM. The device may be defective; try another device.

20 — **Non-blank device** — Appears after the programmer has performed a blank check on a device and has detected bits that are not in their erased or blank state, and are not illegal bits. This error is the result of either the **B** (Blank Check) command or a **P** (Program) command with the blank check option set previously by the extended command **2A** (Enable Programming Option).

21 — **Illegal bit error** — Appears when the programmer has detected a device that has a bit programmed to the incorrect state. When this error code appears, try erasing the part (if possible) and then attempt to program the part again. (This error indicates a fatal condition in non-erasable devices.) If this error code continues to appear, it may be because the device is defective. Discard the part and try another device. The illegal-bit check error occurs as the result of either a **T** (Illegal Bit Check) command or a **P** (Program) command with the illegal bit option set previously by the extended command ***hhh*2A]** (Enable Programming Option).

22 — **Device programming error** — Appears when the programmer detected a defective memory cell in a device during the programming operation. If this error code appears, try another device.

Note: The two following errors have the same error code. For either error to appear, you must have selected command ***n*23]** *(Select Verify Option). If* **1** *was specified as the variable, use the first description. If* **2** *was specified, use the second description.*

23 — **Verify data error (**V_{CC} Nominal) — Appears when the programmer has performed a Verify and has found a memory cell that was not programmed correctly. The device was verified while being operated with its normal operating voltage applied. When this error code appears, try another device.

23 — **Verify data error (**V_{CC} low) — Appears when the programmer has performed a Verify and has found a memory cell that was programmed incorrectly. The device was verified while being operated with its lowest operating voltage applied. When this error code appears, attempt to program the device again. If this error code reappears, try a different device.

24 **Verify data error (**V_{CC} high) — Appears when the programmer has performed a Verify and has found a memory cell that was programmed incorrectly. The device was verified while being operated with its high operating voltage applied. When this error code appears, attempt to program the device again. If this error code reappears, try a different device.

27 **End of user RAM exceeded** — There is not enough user RAM for the amount of data you want to load into it or program from it. You may have the device block size set too large, or the beginning RAM address too high. The operation can still be performed, but only part of the device will be programmed.

28 **Fatal device-specific programming error** — Generic error code generated by the device-specific algorithm. No further commands should be attempted until the error condition is corrected.

29 **Non-fatal device-specific programming error** — Generic error code generated by the device-specific algorithm. This error is non-fatal and is for information purposes only. In most cases, the operation was performed successfully.

2A **Device Insertion error** — Appears when either the device socket is not in the locked position, or the device is inserted backwards, or is not bottom-justified in the socket, or the device pins are not making good contact. Check the device's continuity in the socket and then try the operation again. If the same error code appears, try a different device.

Note: The two following errors have the same error code. For either error to appear, you must have previously selected command ***n*****23***, where n = 0, 1, or 2. If a 0 is chosen, there will be no error condition. (Select Verify Option.) If* **1** *was specified as the variable, use the first description. If* **2** *was specified, use the second description.*

2B **Structured test error (**V_{CC} Nominal) — Appears when the programmer has performed a functional test on a logic device and has detected a failure. If you had selected **1** as the number of Verify Passes, the programmer was attempting to verify the logic device at its normal operating voltage. When this error code appears, try another device.

2B **Structured test error (**V_{CC} low) — Appears when the programmer has performed a functional test on a logic device at the low voltage and has detected a failure. If you had selected **2** for the number of Verification passes, one pass is performed while the lowest specified operating voltage is applied to the device, and the second pass is performed while the highest specified operating voltage is applied to the device. When this error code appears, try another device.

2C **Structured test error (**V_{CC} high) — Appears when the programmer has performed a functional test of a logic device at the high voltage and detected a failure. If you had selected **2** for the number of Verification passes, one pass is performed while the lowest specified operating voltage is applied to the device, and the second pass is performed while the highest specified operating voltage is applied to the device. When this error code appears, try another device.

2D	**Base/Adapter for device not installed** — The device you selected cannot be programmed in the Base that is presently installed. If you select a device that is not supported by the installed Base, no device operations can be performed. If you select a device that uses the PPI Base, make sure the correct adapter is installed. Install the correct Base/Adapter and try again.
2E	**Programming hardware hasn't passed self-test** — Occurs when a programming operation is attempted and the self-test had previously failed for critical hardware. Return to local mode and check the self-test screen.
30	**Device algorithm not found** — Occurs if device selection is attempted using family/pinout codes, and the codes selected correspond to an algorithm which is not supported by the programmer. This error code will also occur when a device operation is attempted prior to having a device selected.
31	**Device over-current fault** — Appears when you attempt to program a socketed device whose programming current is higher than the device you selected on-screen. The device may be faulty; insert another device into the socket and try the operation again.
3B	**No device present** — Appears when a device operation was attempted with no device in the socket.
40	**I/O initialization error** — Appears after an attempt to initialize the Remote port has failed. Check connections and attempt the operation again.
41	**Serial-framing error** — Appears when the Remote port detects a start bit, but the stop bit is positioned incorrectly. Check the baud rate and stop bit setting for the Remote port, or use hardware handshaking.
42	**Serial-overrun error** — The Remote port received characters that the programmer was unable to service. Check the baud rate and stop bit settings for the Remote port, or use hardware handshaking.
43	**Serial framing/overrun error** — This is a combination of serial-framing error 41 and overrun error 42. Check the baud rate and stop bit settings for the Remote port,.or use hardware handshaking.
46	**I/O timeout** — Too much time passed before the programmer received a data file during a download operation. The I/O timeout period may be changed with the Select I/O Timeout Command (=).
52	**Data verify error** — The data from the Remote port did not match the data in RAM. Check the data and try the operation again.
75	**Security Fuse Violation** — Appears if you tried to load, program, or verify data from a device that has its security fuse programmed. If this error code appears, use a master device whose security fuse is still intact.
77	**Security fuse programming error** — Appears if the programmer cannot program the security fuse. If this error code appears, the device you are trying to program may be defective; try programming another device.

79	**Preload not supported by this device** — Appears when a preload vector in the programming data cannot be applied to the logic device.
81	**Serial-parity error** — The Remote port detected incoming data that had incorrect parity. Check the parity setting for the Remote port.
82	**Sumcheck error** — The sumcheck of the data received (as the result of a download) did not match the sumcheck downloaded from the host computer. The host computer sends a transmission sumcheck, and possibly a fuse map sumcheck, as a part of the data record. The programmer compares those sumchecks with the sumchecks it created on that same data. If the two sumchecks do not match, this error code will appear, indicating that some of the data transmitted by the host was not received by the programmer. Try the operation again, and if the problem continues, verify that the sumchecks generated from the host are correct. If the sumchecks are correct, contact Data I/O Customer Support.
84	**I/O format error** — There is a compatibility problem with the data translation format you are using. Check the format of the data. The Translation Formats chapter of this manual contains a description of all the data translation formats supported by the programmer. Or try sending a different translator format. If format 04 is selected, this error can indicate an illegal parameter error. Since this format is a word oriented format, the following parameters must be set to even values; I/O offset, memory begin address, user data size and upload record size.
88	**Invalid number of parameters** — Appears when a CRC command is preceded by an invalid number of parameters. Check the parameters preceding the command and re-issue the command.
89	**Illegal parameter or invalid operation selected** — Appears if an illegal parameter or option precedes a CRC command. For example, this code is returned if you setting the number of verify passes to 0 and then performing a verify. This message also appears if you attempt an operation that is not compatible with the selected device. For example, this code is returned if you attempt a byte swap with a logic device selected.
8B	**Error restoring/saving CRC user-defined parameters or restoring CRC entry defaults.** — An error occurred while attempting to restore or save CRC user-defined parameters (commands **FD** or **FE**) or restoring CRC entry defaults (command **FC**). Check to make sure the disk is not write-protected and that the Algorithm/System disk is in the disk drive.
8E	**File error** — A disk file error occurred during a command that accesses a disk file, such as Load File from Disk, Yield Tally, or Select Device. If you are loading a file from disk, the file probably does not exist, or the filename is misspelled. Check the filename. If you are doing a Yield Tally or selecting a device, the disk is probably write-protected. Remove the write-protection and try the operation again. This error code also occurs when you are performing a Save Configuration operation and the Algorithm Save area is exceeded.

8F	**NON-JEDEC data present in RAM or disk file, or else a NON-logic device was selected with a JEDEC I/O translation format selected** — If you are doing a device operation or a load file from disk operation, check your JEDEC data. If you are doing an upload operation, select the logic device for the JEDEC file to be uploaded, or select a different I/O format for a memory device.
90	**Illegal I/O format** — You tried to select an I/O format that is not supported by the programmer or you attempted to select a non-JEDEC format when a logic device was selected. Select a valid format. See the Translation Formats section for a list of supported formats.
94	**Data record error** — The data that you attempted to transfer did not conform with the selected translation format; edit the data file so that it matches one of the programmer's supported translator formats. See the Translation Formats section for output samples of each translator.
97	**Block move error** — A block move within RAM has violated the RAM boundaries. Check the memory begin address and memory block size and try the operation again.
98	**End of device exceeded** — There is not enough room in the device to hold all the data you have specified. You may have the device beginning address set too high, the block size set too high, or you may need a larger device. Although the operation may still be performed, only part of the data will be programmed into the device.
99	**End of file exceeded** — Appears when the memory block size and memory begin address parameters you specified in the Programming screen are too large for the data file you intend to use for programming. When this error code appears, change the memory block size and memory begin address file size parameters so they are small enough to accommodate the data file. You can perform the operation without changing anything, but only part of the device will be programmed.
9A	**Algorithm disk cannot be found** — Appears if you are selecting a device and do not have the Algorithm/System disk installed. Insert the Algorithm/System disk in the drive and send the device selection command again.
9B	**Incompatible system/algorithm revision numbers or Missing device libraries.** — This error code is returned when the version number of the algorithm file is not compatible with the system software during a device selection operation. Insert the disk with the correct algorithm file. This error code can also be returned as a result of a device selection operation using the 3E] command when the device being selected requires a device library which is not currently installed on the programmer. In order to correct this condition you must purchase the appropriate device library from Data I/O.

9D — **I/O address beyond range of data format selected** — An I/O address exceeded the highest value allowed in the address field of the data format selected. Prior to performing an upload or output to disk operation the programmer calculates the highest I/O address that will be output based on the parameters supplied by the user and aborts the operation if the I/O address is too large for the data format selected. The formula for calculating the highest I/O address is:

Highest I/O address = I/O addr offset + User data size - 1

To avoid this error condition either select a different data format (one which supports the I/O addresses for the transfer operation) or decrease the value of the I/O offset address and /or the User data size to achieve I/O addresses within the range of the data format selected. The I/O addr offset parameter is considered an unsigned value. If it is set to the special default value of FFFFFFFF, it is treated as a value of 0. See the section titled "Table of Highest I/O Addresses" in Chapter 7 for more a list of the highest I/O address allowed for each translation format.

A1 — **No Electronic ID** — The device does not contain an electronic ID. Turn off the Electronic ID option or change devices.

A2 — **Electronic ID verify error** — The device you tried to program did not have the correct electronic ID. Insert the correct device in the socket, or select a different device.

AB — **Unable to load system file from system disk** — You tried to exit or suspend CRC and the Algorithm/System disk was not in the disk drive. Make sure the Algorithm/System disk is in the drive when you exit or suspend CRC.

AC — **Security violation** — You tried to use a new version of system software that has not been installed yet. Exit CRC mode and perform the Update command to install the new version of software.

AE — **Keep Current algorithm disk not found. Insert your Keep Current algorithm disk** — This error code is returned when the Keep Current algorithm file for the specified device is not found. Insert the disk with the Keep Current algorithm file for the specified device, and try again.

AF — **Operation not allowed because the device was selected by family/pinout code** — This error code is returned when you attempt certain device operations, such as Compare Electronic ID, after having used a family/pinout code (the CRC @ command) to select a device. Use the *xxx...xxxx*33], *xxx...xxxx*34], and *n*40] commands to reselect the device using device manufacturer and device part number.

To avoid this error, rewrite your CRC driver to use the device manufacturer and device part number to select devices.

B1 — **Block not allowed for bulk erase** — This error is returned when you try to bulk erase part of a device and the device does not support partial bulk erasing. If you want to bulk erase the device, you will have to erase the entire device. To bulk erase the entire device, set the Device Begin Address to 0 and the Memory Block Size to the size of the device to 0.

B2 Partial device operation not allowed — This message appears when the attempt to program a device fails because of one or both of the following reasons:

- The selected device must be programmed in its entirety when a programming operation is used on it. The last programming operation attempted to program only a portion of the device.
- The User Data is incorrect for the the selected device.

To make sure the entire device is programmed, set both the **Device Begin Address** and the **Device Block Size** to **0** (when Device Block Size is set to 0, the system sets the Device Block Size to the size of the entire device(s)).

To make sure the **User Data** is correct in the device operation, enter **0** in the **User Data Size** field (the system sets the User Data Size to cover the entire device(s)).

B4 **Odd Memory Begin Address or User Data Size Incompatible with Data Word Width** — This error code is returned when you try a device operation on a 16-bit (or larger) device and either the Memory Begin Address is set to an odd number or the User Data Size is not compatible with the Data Word Width selected. Frequently, this happens when a 16-bit device is used and the User Data Size (defined in bytes) is an odd byte count. Adjust your User Data Size or the Memory Begin Address.

BC **Last device operation checksum not available.** — This error is returned when no successful load, verify, or program operation has been done since the programmer was last powered up, or when the last load, verify, or program operation was unsuccessful.

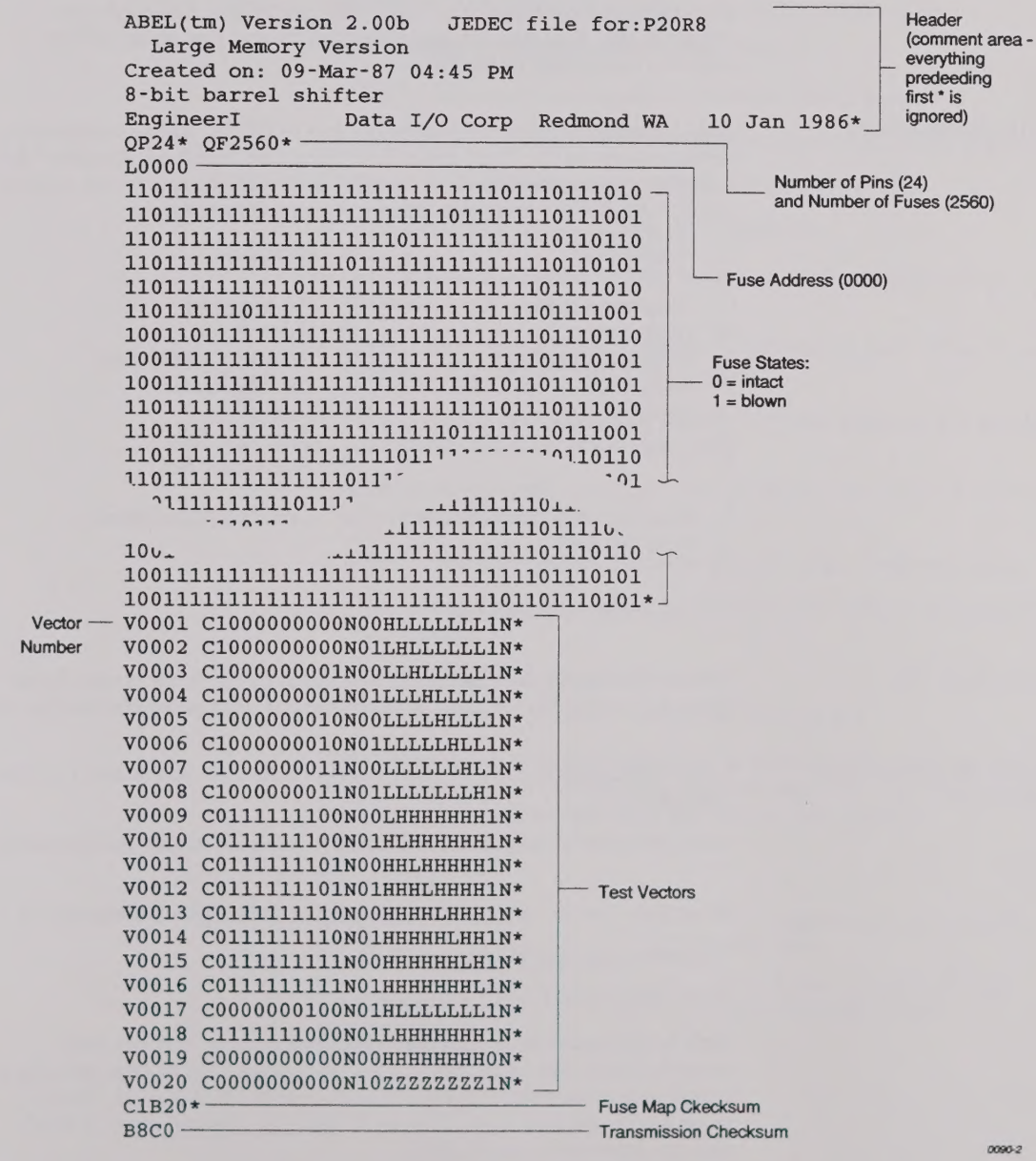
D5 **Port Transfer Error** — Appears when you try to transfer data over a serial port which is not properly connected. If this error occurred after using the ***n*3C]** command to set the data port to the Terminal port, ensure that the Terminal port is properly connected.

FE **Undefined error** — An error occurred that the CRC program could not categorize. Document the method in which the error occurred and call Data I/O to report the problem.

JEDEC Full Format, Code 91

The "full" JEDEC format consists of a start-of-text character (STX), various fields, an end-of-text character (ETX), and a transmission checksum. A sample JEDEC transmission sent in the "full" format is shown in Figure 7-18. Each of the fields is described on the following pages.

Figure 7-18
An Example of JEDEC Full Format



JEDEC Field Syntax

<field> ::= [<delimiter>]<field identifier>{<field character>}'*'

<field identifier>::= 'A' | 'C' | 'D' | 'F' | 'G' | 'K' | 'L' | 'N' | 'P' | 'Q' | 'R' | 'S' | 'T' | 'V' | 'X'

<reserved identifier>::= 'B' | 'E' | 'H' | 'I' | 'J' | 'M' | 'O' | 'U' | 'W' | 'Y' | 'Z'

Following the design specification field in a JEDEC transmission can be any number of information fields. Each of the JEDEC fields begins with a character that identifies what type of field it is. Fields are terminated by using an asterisk character. Multiple character identifiers can be used to create sub-fields (i.e., "A1", "A$", or "AB3"). Although not required, you may use carriage returns (CR) and line feeds (LF) to improve readability of the data.

Field Identifiers

Field identifiers which are currently used in JEDEC transmissions are shown above on the "field identifiers" line. The "reserved identifier" line indicates characters not currently used (reserved for future use as field identifiers). JEDEC field identifiers are defined as follows:

A	Access time	N	Note field
B	*	O	*
C	Checksum field	P	Pin sequence
D	Device type	Q*x*	Value field
E	Electrical data field	R	Resulting vector field
F	Default fuse state field	S	Starting vector
G	Security fuse field	T	Test cycles
H	*	U	User data field
I	*	V	Test vector field
J	*	W	*
K	Fuse list field (hex format)	X	default test condition
L	Fuse list field	Y	*
M	*	Z	*

* Reserved for future use

Device Field (D)

Device selection by this field is not supported by the programmer. It has been replaced by the QF and QP fields and the manual selection of devices.

Fuse Information Fields (L, K, F, C)

<fuse information> :: = [<default state>] <fuse list> {<fuse list>} [<fuse checksum>]

<fuse list> :: = 'L' <number> <delimiter> {<binary-digit> [<delimiter>]} ' * '

<fuse list> :: = 'K' <number> <delimiter> {<hex-digit> [<delimiter>]} '*'

<default state> :: = 'F' <binary-digit> ' * '

<fuse checksum> :: = 'C' <hex-digit>:4 ' * '

Each fuse of a device is assigned a decimal number and has two possible states: zero, specifying a low-resistance link, or one, specifying a high resistance link. The state of each fuse in the device is given by three fields: the fuse list (L field or K field), the default state (F field), and the fuse checksum (C field).

JEDEC U and E Fields

AutoSite supports the optional JEDEC U (user data) and E (electrical data) fields. The U and E fields are described below.

Note: Implementation of the JEDEC U and E fields is not part of the JEDEC-3C (JESD3-C) standard.

User Data (U Field)

The U field allows user data fuses that do not affect the logical or electrical functionality of the device to be specified in JEDEC files. For instance, the U field can be used to specify the User Data Signature fuse available in some types of PLD devices because this fuse contains information only (it has no logical or electrical functionality).

The following guidelines apply to the U field:

- The U field must be included for devices with U fuses.
- Each U-field cell must be explicitly provided if the U field is present.
- The F (default fuse state) field does not affect U fuses.
- There can only be one U field in a JEDEC file.
- The U field fuses must be listed in the order they appear in the device.
- The U field must be listed after the L field and E fields (if used), and before the V (test vector) field (if used).
- The U field is specified using binary numbers, since the full number of U-field cells is otherwise unknown.
- The number of cells specified in the U field is not included in the QF (number of fuses) field.
- The U-field cells are not included in the C (fuse checksum) field.
- The U field reads left to right to be consistent with the L (fuse list) and E fields.

The syntax for the U field is as follows:

```
<User Data Fuse List>::'U'<binary-digit(s)>'*'
```

The character U begins the U field and is followed by one binary digit for each U fuse. Each binary digit indicates one of two possible states (zero, specifying a low-resistance link, or one, specifying a high-resistance link) for each fuse.

For example,

```
QF24*
L0000
10101100000000000000000*
E10100111*
C011A*
U10110110*
```

The U field must be specified after the L field.

If the E field is used, the U field must be specified after it.

Electrical Data (E field)

The E field allows special feature fuses that do not affect the logic function of the device to be specified in JEDEC files.

The following guidelines apply to the E field:

- Each E-field cell must be explicitly provided if the E field is present.
- The F (default fuse state) field does not affect E fuses.
- There can only be one E field in a JEDEC file.
- The E field fuses must be listed in the order they appear in the device.
- The E field must be listed before the C (checksum) field. If the U field is used, the E field must come before the U (user data) field.
- The E field is specified using binary numbers, since the full number of E-field cells is otherwise unknown.
- The number of cells specified in the E field is not included in the QF (number of fuses) field.
- The E-field cells are included in the C (fuse checksum) field.
- The E field reads left to right for the purpose of checksum calculation.

The syntax for the E field is as follows:

```
<Electrical Data Fuse List>::'E'<binary digit(s)>'*'
```

The character E begins the E field and is followed by one binary digit for each E fuse. Each binary digit indicates one of two possible states (zero, specifying a low-resistance link, or one, specifying a high-resistance link) for each fuse. For example,

```
QF24*
L0000
10101100000000000000000*
E10100111*
C011A*
U10110110*
```

The E field must come before the C field.

If the U field is used, the E field must be specified before it.